UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/770,705	01/26/2001	Christopher S. Gouge	MS155721.2	6801

27195 7590 06/28/2005

AMIN & TUROCY, LLP
24TH FLOOR, NATIONAL CITY CENTER
1900 EAST NINTH STREET
CLEVELAND, OH 44114

EXAMINER	
INGBERG, TODD D	

ART UNIT	PAPER NUMBER
2193	

DATE MAILED: 06/28/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/770,705	GOUGE ET AL.	
	Examiner	Art Unit	
	Todd Ingberg	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *21 April 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-24* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-24* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *26 January 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Claims 1 – 24 have been examined.

Claims 1, 9, 13, 17, 22, 23 and 24 have been amended.

Claim Rejections - 35 USC § 101

1. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claim 23 is rejected under 35 U.S.C. 101 because the claimed invention lacks patentable utility.

A communications signal transmitting between two computers a data packet is not tangible.
Claim 23
A data packet transmitted as a communication signal between at least two computer processes, comprising: a configurable module having: one or more configurable data elements, wherein one or more default values for the one or more configurable data elements are available; one or more non-configurable data elements describing the one or more configurable data elements; and one or more transformation instructions that facilitate configuring the one or more configurable data elements, wherein the instructions are employed to facilitate installation of the one or more configurable data elements into a target data set residing in at least one of the at least two computer processes.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claims 1 – 24 are rejected under 35 U.S.C. 102(b) as being anticipated by Microsoft's Visual C++ version 5.0 as documented in the text book, "Beginning Visual C++ 5", by Ivor Horton, published March 19, 1997. Referred to as **DLL**.

Term's in the Art

The following are terms in art the one of ordinary skill in the art should have knowledge of at the time of invention and the interpretations given during prosecution.

A. Dynamic-Link Library (DLL) - [Microsoft Computer Dictionary, page 166, published September 19, 1997] A feature of the Microsoft Windows family of operating systems and OS/2 that allows executable routines to be stored separately as files with the DLL extensions and to be loaded only when needed by a program. A dynamic-link library has several advantages. First, it does not consume any memory until it is used. Second, because a dynamic-link library is a separate file, a programmer can make corrections or improvement s to only that module without affecting the operation of the calling program or any other dynamic-link library. Finally, a programmer can use the same dynamic-link library with other programs.

B. Dynamic Link Library (DLL) as defined by the IBM Dictionary page 225. "A file containing executable code and data bound to a program at load time, or run time, rather than during linking. The code and data in a dynamic link library can be shared by several applications simultaneously."

Claim 1

DLL anticipates a system that facilitates software installation comprising: a transformation component that receives one or more configurable data elements, and one or more non-configurable data elements describing the one or more configurable data elements; and a merge component that employs one or more transformation instructions that configures the one or more configurable data elements to facilitate the installation of the one or more configurable data elements into at least one target data set.

Examiner's Response

By definition as provided in the section Term's in the Art, a DLL adds data and/or executable code at link or runtime and as the name states the file is linked. The data elements are the different portions of the program which the data elements or functions may or may not be updated replaced by the linkages to the DLL. DLL on page 719 figure shows a library of DLLs and the program which only a portion is updated with the linkages to the DLL replacing the function. The functions in the program correlate to the functions in the DLL and the data elements relate to data in the DLL. the transformation is the runtime update of the linkages causing the installation of the software to transform the runtime environment. DLL teaches many uses of DLLs from pages 715 to 734. Pages 715 to 716 provide an overview of DLLs, page 717 to 719 teach How DLLs Work 720 to 722 teach the content of a DLL, pages 722 – 729 show the tool for performing the task. Pages 729 to 733 clearly ties into the definition by examples.

Claim 2

The system of claim 1, wherein the one or more configurable data elements are stored in a data structure associated with the configurable module.

Examiner's Response

The DLL is a file files inherently have a data structure in view of the rejection for claim 1.

Claim 3

The system of claim 2, wherein the data structure is a metadata item description table.

Examiner's Response

The metadata (data about data by definition) is the address of the linkages which control the running of the program. Most college text books refer to this as a program look up table. the presence is seen in the rejection for claim one with the linkages. The affect of the DLL on the metadata is the linkages instruct the merging of components to perform replacements as illustrated in the rejection for claim 1.

Claim 4

The system of claim 1, wherein the one or more non-configurable data elements are stored in a data structure associated with the configurable module.

Examiner's Response

A non configurable data element is a portion of the program not changed by the DLL as described in the rejection for claim 1.

Claim 5

The system of claim 4, wherein the one or more non-configurable data elements are stored in a metadata item description table.

Examiner's Response

The portions of the program not changed by the DLL inherently are stored in the program lookup table (metadata).

Claim 6

The system of claim 1, wherein the one or more transformation instructions are stored in a data structure associated with the configurable module.

Examiner's Response

The actual transformation instructions are the linkages which are addresses to the beginning and end of functions.

Claim 7

The system of claim 6, wherein the one or more transformation instructions are stored in a transformation instruction table.

Examiner's Response

The presence of a function to be replaced in the figure on 719 is evident of at least one transformation.

Claim 8

The system of claim 2 wherein the data structure includes at least one of a name of a configurable data element, and a semantic meaning for the configurable data element.

Examiner's Response

The name is the name of the function being replaced in the example in claim 1. The semantic meaning being the linking of executable code and/or data at link or runtime must be compatible in order to run. This is deemed inherent.

Claim 9

DLL anticipates a data interpretation system, comprising: a data interpretation component that receives one or more configurable data elements from a configurable module
wherein the one or more data elements include configuration information related to installing the one or more data elements into a software program; applies one or more transformation instructions from the configurable module to the configurable data elements to configure the configurable data elements, and installs the configurable data elements into a target data set based at least in part upon the configuration information,

Examiner's Response

As per claim 1 the definition of a DLL is the ability to add data and/or executable code at link or runtime. The example in claim 1 is of a code replacement. The mechanism to change the address to a data element is the same and is inherent in the rejection of claim 1 by definition. Page 729 to 730, adding variables from a DLL is covered.

Claim 10

The system of claim 9, further comprising a user interface to enable a user to query the configurable module to determine which of the one or more data elements are configurable.

Examiner's Response

Page 725 shows the user interface to create DLLs and search.

Claim 11

The system of claim 9 further comprising: a merging component adapted to receive one or more updated configurable data elements from the data interpretation component and adapted to provide the one or more updated configurable data elements to a target data set.

Examiner's Response

As per the rejection for claim 1.

Claim 12

The system of claim 9 further comprising: an authoring schema that describes a configurable module.

Examiner's Response

DLL Chapter 18 is dedicated to being able to write your own DLLs. Page 717 to 719 explain How DLLs Work.

Claim 13

DLL anticipates a method for installing a configurable data set into a target data set, comprising:

obtaining one or more data elements from a configurable module; including metadata that describes the configuration options of the configurable data set obtaining one or more transformation instructions from the configurable module; and applying the one or more transformation instructions to a copy of the one or more data elements from the configurable module to configure the one or more data elements for installation into the target data set.

Examiner's Response

The rejection for claim 1 covers the limitations for claim 13.

Claim 14

The method of claim 13, further comprising: identifying a target data set; and inserting the updated data elements into the target data set.

Examiner's Response

DLL pages 729 to 730 show adding a variable.

Claim 15

The method of claim 13 further comprising: presenting one or more configuration options to a user; accepting one or more configuration selections from the user; and selectively configuring the one or more data elements based on the user's configuration selections.

Examiner's Response

Page 728 shows several configuration options in the figure link incrementally, ignore all default libraries etc.

Claim 16

A computer readable medium containing computer executable instructions operable to perform the method of claim 13.

Examiner's Response

DLL by definition.

Claim 17

DLL anticipates a method for creating a configurable data module, comprising: creating a configurable data set having one or more configurable data elements; and creating one or more data structures containing information associated with one or more configurable data elements, and displaying the information to a software program to facilitate installing the configurable data set into the software program.

Examiner's Response

The rejection for claim explains the details of DLLs and page 725 shows a user interface. Also see page 727 the Sketcher program for determining the addresses for linkage.

Claim 18

The method of claim 17 wherein creating a configurable data set includes: identifying one or more attributes of the one or more data elements; and establishing one or more default values for the attributes of the one or more data elements.

Examiner's Response

DLL, page 724 top of the page ExtDLLExample.cpp (key is the .cpp) contains the information.

Claim 19
The method of claim 17 wherein creating the one or more data structures further comprises: identifying one or more locations within a data set that are configurable; identifying one or more configuration options; creating one or more instructions concerning how to configure the one or more locations; and storing the instructions in the one or more data structures.
Examiner's Response
DLL, page 724 top of the page ExtDLLExample.def (key is the .def) contains the information.

Claim 20
The method of claim 19, wherein the one or more data structures are stored in the configurable data module.
Examiner's Response
DLL, page 724 the file names at the top of the page.

Claim 21
A computer readable medium containing computer executable instructions operable to perform the method of claim 17.
Examiner's Response
DLL by definition.

Claim 22
DLL anticipates a system for installing a configurable data set into a target data set, comprising: a configurable module having configurable data elements representing a configurable data set and non-configurable data elements representing a portion of the configurable data set; a user interface for selecting which configurable data element to modify; and a data interpretation system for receiving the configurable and non-configurable data elements from the configurable module and applying the transformation instructions applicable to the user selections associated with the configurable data elements to enable installing the configurable data set into to the target data set.
Examiner's Response
See the rejection for claim 1.

Claim 23
DLL anticipates a data packet adapted to be transmitted between at least two computer processes, comprising: a configurable module having: one or more configurable data elements, wherein one or more default values for the one or more configurable data elements are available; one or more non-configurable data elements describing the one or more configurable data elements; and one or more transformation instructions that facilitate configuring the one or more configurable data elements, wherein the instructions are employed to facilitate installation of the one or more configurable data elements into a target data set.
Examiner's Response
See the rejection for claim 1.

Claim 24
DLL anticipates a computer readable medium having stored thereon a data structure, comprising: a first data field containing one or more configurable data elements, wherein one or more default values for the one or more configurable data elements are available; a second data field containing one or more non-configurable data elements describing the one or more configurable data elements; and a third data field containing one or more transformation instructions that facilitate configuring the one or more configurable data elements, to load the configurable data elements into a software program.
Examiner's Response
The rejection for claim 1 teaches the details of DLLs. Default values can be interpreted the linkages or possibly the importing of Symbols page 730 to 733.

Response to Arguments

4. Applicant's arguments filed April 24, 2005 have been fully considered but they are not persuasive. Arguments have been scanned and may contain some errors as a result from the scanning.

Rejection of Claims 1-24 Under 35 U.S.C. &102(b)
Applicant's Argument
"Applicants' claimed invention relates to a system and method for creating and describing a configurable merge module wherein a data set in the configurable merge module is configured, and the resultant: configured data set is merged into a target data set. In particular, the invention as claimed relates to creating, describing and configuring software components (that are data sets) to be incorporated into software programs (target data sets) such that the software components are self-describing in relation to configuration possibilities for the particular software program. More particularly, independent claims 1, 9, 13, 17, 22, 23 and 24 recite similar claim limitations, namely: ... Installation of the one or more configurable data elements into at least one target data set. Horton does not disclose this novel aspect of applicants' claimed invention."
Examiner's Response
Applicant's claim language does not ***clearly and concisely*** distinguish the invention as a limited to a non runtime environment. The use of Dynamic Link Libraries (DLLS) reads on the claims.

Configurable Merge Module
DLL is defined above as "A file containing executable code and data bound to a program at load time, or run time, rather than during linking. The code and data in a dynamic link library can be shared by several applications simultaneously."(As defined above).

Data Set in the Configurable Merge Module

The content of DLL is configured as designed by the Programmer. DLL by definition being "A file containing executable code and data bound to a program at load time, or run time, rather than during linking. The code and data in a dynamic link library can be shared by several applications simultaneously."(As defined above). The actual "code and data" in the DLL is the data set.

Resultant
The end result of the merge of the content of the DLL (software AND/OR data) is the content of the DLL is merged into a target data set (the software component(s)) program which the content is being modified.

Applicant's claim language for independent claims 1, 9, 13, 17, 22, 23 and 24 fail to distinguish the claimed invention over prior art of record. The well known uses of DLLs perform installation of the one or more configurable data elements into at least one target data set.

In addition, to limitations regarding non load time or runtime (i.e. development environment version management, managing source code etc) the Applicant may considered specifying the form of what is being merged. Software or Data is not distinct. Merging "source code" is distinct. Source code has not yet been compiled. Please, take careful note of the Assignee's reference the Examiner made of record. The commercial product Visual Source Safe (VSS). The Examiner specifically mentioned this reference in anticipation the Applicant would claim the intended environment. VSS is mentioned in the last office action and the merge feature was pointed out.

Applicant's Argument

"The Office Action asserts that Norton provides substantiation for the rejection of the subject claims under 35 U.S.C. § 102. Applicants' representative respectfully disagrees. Horton discloses the creation and utilization of dynamic link libraries, and in particular that a dynamic link library :is a file containing a collection of modules that can be used by any number of different programs. However, Horton does not teach or suggest the installation of one or more: configurable data elements into at least one target data set, thereby transforming the at least one target data set with configurable data elements. Moreover, Horton reiterates this point by stating at page 718 that: "[no] code from a DLL is included [installed] in the executable module of any of the programs." In contrast, the invention as claimed installs and/or loads configurable data elements into target data sets through the use of transformation instructions applied to configurable data elements to effectively modify/transform the target data set with the configurable elements obtained from a transformation component."

Examiner's Response

As to Applicant's statement "Horton discloses the creation and utilization of dynamic link libraries, and in particular that a dynamic link library : is a file containing a collection of modules that can be used by any number of different programs." In view of the description above of the limitations not clearly and concisely distinguishing the invention. Furthermore, no present claim limitation make the ability to access the DLL from more than one program a distinguishing feature. Also, the DLL has the ability to be a collection of modules and can be used by any number of different programs. This is not a technical limitation, requiring multiple or a plurality.

It simply states it can support multiple. In the same sense it does not limit the use with one program or providing only one module to merge with a program.

NOTE: Applicant's argument that the invention is distinct because it can not support made it can not support a plurality or multiples is noted for the purpose of file wrapper estoppel.

As to Applicant's statement "However, Horton does not teach or suggest the installation of one or more: configurable data elements into at least one target data set, thereby transforming the at least one target data set with configurable data elements." In reviewing the Examiner's response to the first argument the Examiner disagrees. Furthermore, the overview is provided in response to the first argument.

As to Applicant's statement" Moreover, Horton reiterates this point by stating at page 718 that: "[no] code from a DLL is included [installed] in the executable module of any of the programs." In contrast, the invention as claimed installs and/or loads configurable data elements into target data sets through the use of transformation instructions applied to configurable data elements to effectively modify/transform the target data set with the configurable elements obtained from a transformation component."." The Applicant fails to provide limitations that clearly and concisely claim the code is "source code". DLLs by definition do in fact install code or data into a target data set as described above. DLLs transformation instructions are a means of updating pointers to link the updates.

Applicant's Argument

"In addition, applicants' representative contends that the Office Action misconstrues and mischaracterizes the target data set as claimed as being equivalent to a runtime environment. Target data sets as provided in the subject application are software programs, (See page 2, lines 22-23). A runtime environment in contrast is a transient state wherein the utilization of a DLL has no transformative effect on the underlying invoking software program, when the runtime environment has terminated, the program(s) invoking the DLL as well as the DLL itself, are left in exactly the same condition that they were in prior to execution and invocation of the DLL unchanged/untransformcd. The invention as claimed on the other hand forces a transformation such that after utilization of the claimed invention the resultant modified target data set is no longer the entity that it was prior to the: utilization of the claimed invention."

Examiner's Response

As stated previously, Applicant failed to clearly and concisely claim the software programs are source code. In fact, software programs can also mean executable programs. Applicant's argument that the DLL is in the same state and has not changed is accurate. The Applicant also acknowledges the linkages below. What Applicant is not acknowledging is the target data set is modifier producing the resultant. In basic terms, the DLL links to the target modifying the content of that target producing an altered software program Applicant calls the end result the resultant.

Applicant's Argument

"The Office Action farther appears to mischaracterize applicants' claimed invention as merely manipulating and updating linkages between a DLL and a program that invokes the DLL. As has been stated supra, the invention as claimed incorporates/installs software components

(that are configurable data elements) into software programs (that are target data sets), such that by the act of incorporating/installing software components into software programs, a software program i5 transformed from being a software program without a particular software component to being a software program with a particular software component incorporated/installed therein. Thus, the invention as claimed rather than mere] y manipulating and updating linkages between an external DLL and an invoking program such that the external DLL and invoking program remain separate and distinct, incorporate & installs software components into the body of the software program thereby rendering a unitary entity."

Examiner's Response

The majority of the arguments have been answered above. As to the Applicant's statement that the result is a "unitary entity". This limitation with the clear and concise limitations that "source code" is being update in a non runtime environment is absent from the claims.

Applicant's Remarks

"Further, the examiner is reminded that the standard by which anticipation is to be adjudged is strict identity between the; cited document and the invention as claimed, riot mere equivalence or similarity. See, Richardson at 9 USPQ2d 1913, 1920. This means that in order to establish anticipation under 35 U.S.C. §102, a single document must not only expressly or inherently describe each and every limitation set forth in the patent claim, but also the identical invention must be shown in as complete detail as is contained in the claim. Lt is submitted that Horton does not provide the necessary identity to substantiate the rejection under 35 U.S.C. §102, and in particular the cited document does not provide for the installation of configurable data elements into a target data set thereby rendering a single indivisible entity a software program."

Examiner's Response

The case law Richardson v. Suzuki Motor Co., 868 F.2d 1226, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989) which mentions identical invention, Applicant relies on as fitting the fact pattern for overcoming the rejection based on case law has been reviewed. In this case, Richardson modified an existing motorcycle suspension. The case involves modifying an existing commercial product. Applicant has not disclosed the existing commercial product they modified to produce the instant invention. Applicant is reminded of their duty to disclose such relevant information under Rule 156 (emphasis added).

Any amendments to the claims where the "non runtime environment" and update of "source code" with the merge limitations present must be supported in the Specification.

Conclusion

5. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Correspondence Information

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Todd Ingberg whose telephone number is (571) 272-3723. The examiner can normally be reached on during the work week..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571) 272-3719. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Todd Ingberg
Primary Examiner
Art Unit 2193

TI